IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

CARDSOFT (ASSIGNMENT FOR THE §
BENEFIT OF CREDITORS), LLC §
§
v. § CASE NO. 2:12-CV-325-JRG-RSP
§
THE GORES GROUP, LLC, et al. §
§

CLAIM CONSTRUCTION MEMORANDUM AND ORDER

On November 20, 2013, the Court held a hearing to determine the proper construction of the disputed claim terms in United States Patents No. 6,934,945 and 7,302,683. After considering the arguments made by the parties at the hearing and in the parties' claim construction briefing (Dkt. Nos. 85, 93, and 97),[1] the Court issues this Claim Construction Memorandum and Order.

[1] Citations to documents (such as the parties' briefs and exhibits) in this Claim Construction Memorandum and Order shall refer to the page numbers of the original documents rather than the page numbers assigned by the Court's electronic docket.

Table of Contents

BACKGROUND

Plaintiff brings suit alleging infringement of United States Patents No. 6,934,945 ("the '945 Patent") and 7,302,683 ("the '683 Patent") (collectively, the "patents-in-suit"). The patents-in-suit are both titled "Method and Apparatus for Controlling Communications" and both bear a priority date in March 1997. The Abstract of the '945 Patent states:

> The present invention relates to preparing and processing information to be communicated via a network or to or from other data carriers. For implementation of a novel "virtual machine" of the present invention, a minimal amount of hardware is required. Prior art virtual machines tend to slow down operation of the device as they interface between an application program and device drivers. The novel virtual machine incorporates a virtual message processing means that is arranged to construct, deconstruct and compare messages and [that is] applied in the native code of the processor. The message instruction means directs and controls the message processor. Similarly, a protocol processor means governs and organs [*sic*, organizes] communications, under the direction of a protocol instruction means in the application. These elements of the novel virtual machine increase the speed and efficiency and allow implementation of a practical device for use in communications, able to be implemented on different hardware having different BIOS/OS.

The Abstract of the '683 Patent states:

> Disclosed is a device arranged to process messages for communications, comprising a virtual machine means including a message processor means which is arranged to process messages communicated to and/or to be communicated from the device, and message processor instruction means, arranged to provide directions for operation of the message processor means. Also disclosed is a method for operating a device arranged to process messages for communications and a method of programming a device arranged to process messages for communications.

The '945 Patent issued on August 23, 2005. The '683 Patent issued on November 27, 2007. The '683 Patent is a continuation of the '945 Patent. Because the patents-in-suit therefore share a common written description and figures, for convenience this Claim Construction Memorandum and Order cites the specification of the '945 Patent unless otherwise indicated.

The Court previously construed claims of the patents-in-suit in *CardSoft (Assignment for the Benefit of Creditors) LLC, et al. v. VeriFone Systems, Inc., et al.*, No. 2:08-CV-98, Dkt. No. 251 (E.D. Tex. Sept. 23, 2011) (Everingham, J.) ("*CardSoft I*").[2] *CardSoft I* proceeded to trial on the merits, and a jury found infringement by defendant Hypercom Corporation ("Hypercom") as to Claim 11 of the '945 Patent and Claim 1 of the '683 Patent. *See* No. 2:08-CV-98, Dkt. No. 389, 6/8/2012 Verdict Form. The Court entered a Judgment as to Hypercom on October 30, 2013. No. 2:08-CV-98, Dkt. No. 483.[3]

LEGAL PRINCIPLES

"It is a 'bedrock principle' of patent law that 'the claims of a patent define the invention to which the patentee is entitled the right to exclude.'" *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312 (Fed. Cir. 2005) (en banc) (quoting *Innova/Pure Water Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). To determine the meaning of the claims, courts start by considering the intrinsic evidence. *See id.* at 1313; *see also C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 861 (Fed. Cir. 2004); *Bell Atl. Network Servs., Inc. v. Covad Commc'ns Group, Inc.*, 262 F.3d 1258, 1267 (Fed. Cir. 2001). The intrinsic evidence includes the claims themselves, the specification, and the prosecution history. *See Phillips*, 415 F.3d at 1314; *C.R. Bard*, 388 F.3d at 861. Courts give claim terms their ordinary and accustomed meaning as understood by one of ordinary skill in the art at the time of the invention in the context of the

[2] The Court herein uses the term "*CardSoft I*" primarily with reference to the *CardSoft I* Memorandum Opinion and Order regarding claim construction (No. 2:08-CV-98, Dkt. No. 251), but in some instances "*CardSoft I*" refers to Civil Action No. 2:08-CV-98 in general. The meaning is evident from the context in which the term is used.

[3] The authorities cited by the parties and otherwise considered by the Court do not readily resolve the collateral estoppel issues in the above-captioned case. On balance, the Court need not resolve whether collateral estoppel extends to Gores Group because upon considering the merits, the Court reaches the same conclusions here as in *CardSoft I* on all issues that were analyzed by *CardSoft I*, as set forth below.

entire patent. *Phillips*, 415 F.3d at 1312-13; *accord Alloc, Inc. v. Int'l Trade Comm'n*, 342 F.3d 1361, 1368 (Fed. Cir. 2003).

The claims themselves provide substantial guidance in determining the meaning of particular claim terms. *Phillips*, 415 F.3d at 1314. First, a term's context in the asserted claim can be very instructive. *Id.* Other asserted or unasserted claims can aid in determining the claim's meaning because claim terms are typically used consistently throughout the patent. *Id.* Differences among the claim terms can also assist in understanding a term's meaning. *Id.* For example, when a dependent claim adds a limitation to an independent claim, it is presumed that the independent claim does not include the limitation. *Id.* at 1314-15.

"[C]laims 'must be read in view of the specification, of which they are a part.'" *Id.* at 1315 (quoting *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 979 (Fed. Cir. 1995) (en banc)). "[T]he specification 'is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.'" *Phillips*, 415 F.3d at 1315 (quoting *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996)); *accord Teleflex, Inc. v. Ficosa N. Am. Corp.*, 299 F.3d 1313, 1325 (Fed. Cir. 2002). This is true because a patentee may define his own terms, give a claim term a different meaning than the term would otherwise possess, or disclaim or disavow the claim scope. *Phillips*, 415 F.3d at 1316. In these situations, the inventor's lexicography governs. *Id.* The specification may also resolve the meaning of ambiguous claim terms "where the ordinary and accustomed meaning of the words used in the claims lack sufficient clarity to permit the scope of the claim to be ascertained from the words alone." *Teleflex*, 299 F.3d at 1325. But, "[a]lthough the specification may aid the court in interpreting the meaning of disputed claim language, particular embodiments and examples appearing in the specification will not generally be read into the claims." *Comark Commc'ns, Inc. v. Harris Corp.*, 156 F.3d 1182, 1187 (Fed. Cir. 1998) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1571 (Fed. Cir. 1988)); *accord Phillips*, 415 F.3d at 1323.

The prosecution history is another tool to supply the proper context for claim construction because a patent applicant may also define a term in prosecuting the patent. *Home Diagnostics, Inc., v. Lifescan, Inc.*, 381 F.3d 1352, 1356 (Fed. Cir. 2004) ("As in the case of the specification, a patent applicant may define a term in prosecuting a patent."). "[T]he prosecution history (or file wrapper) limits the interpretation of claims so as to exclude any interpretation that may have been disclaimed or disavowed during prosecution in order to obtain claim allowance." *Standard Oil Co. v. Am. Cyanamid Co.*, 774 F.2d 448, 452 (Fed. Cir. 1985).

Although extrinsic evidence can be useful, it is "less significant than the intrinsic record in determining the legally operative meaning of claim language." *Phillips*, 415 F.3d at 1317 (citations and internal quotation marks omitted). Technical dictionaries and treatises may help a court understand the underlying technology and the manner in which one skilled in the art might use claim terms, but technical dictionaries and treatises may provide definitions that are too broad or may not be indicative of how the term is used in the patent. *Id.* at 1318. Similarly, expert testimony may aid a court in understanding the underlying technology and determining the particular meaning of a term in the pertinent field, but an expert's conclusory, unsupported assertions as to a term's definition are entirely unhelpful to a court. *Id.* Generally, extrinsic evidence is "less reliable than the patent and its prosecution history in determining how to read claim terms." *Id.*

THE PARTIES' STIPULATED TERMS

The parties have reached agreement on constructions for certain terms, as stated in their August 30, 2013 Joint Claim Construction and Prehearing Statement (Dkt. No. 236 at Ex. A), their briefing, and at the November 20, 2013 hearing. The parties' agreements are set forth in Appendix A to this Claim Construction Memorandum and Order.

CONSTRUCTION OF DISPUTED TERMS

Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with preliminary constructions of the disputed terms with the aim of focusing the parties' arguments and facilitating discussion. Those preliminary constructions are set forth within the discussion of each term, below.

A. "virtual machine means"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"a computer programmed to emulate a hypothetical computer for applications relating to transport of data"	"a computer programmed to emulate a hypothetical computer for *portable* applications relating to transport of data"

Dkt. No. 60, Ex. A at 1 (emphasis added). This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "a computer programmed to emulate a hypothetical computer for applications relating to transport of data."

(1) The Parties' Positions

Plaintiff proposes the *CardSoft I* construction. *See CardSoft I* at 13-14. Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from re-litigating the construction of "virtual machine means." Dkt. No. 85 at 6.

Alternatively, Plaintiff argues that Defendants' proposal of "portable" has no support in the specification or the prosecution history. *Id.* at 7.

Defendants argue that their proposal "clarifies that the Patents-in-Suit are directed to virtual machines that permit the *same* application to run on multiple devices without modification to the device or recertification by the credit card company." Dkt. No. 93 at 17. Further, Defendants argue that Plaintiff should not be allowed to misapply the Court's *CardSoft I* construction:

> It is Defendants' impression that the *CardSoft I* jury was confused as to what structure in the accused products could satisfy the virtual machine means limitation. Defendants understand that CardSoft's expert, Mr. Cole, testified that computer code in source code form—before it is loaded onto a device—was the accused virtual machine means. That is not a reasonable conclusion based on the Court's claim construction, which defines the metes and bounds of the claim term, and such arguments should not be permitted to avoid jury confusion.

Id.

At the November 20, 2013 hearing, Defendants further urged that applications are portable by virtue of the presence of a virtual machine. In other words, Defendants argued, the nature of using a virtual machine is that applications for the virtual machine are necessarily portable. Plaintiff responded that Defendants' proposal should be rejected because Defendants interpret the word "portable" too narrowly. Plaintiff explained that, as a practical matter, some degree of modification may be necessary to accommodate different devices despite the use of a virtual machine.

(2) Analysis

Claim 1 of the '945 Patent is representative and recites (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a *virtual machine means* which includes
>
> a virtual function processor and function processor instructions for controlling operation of the device, and

> message induction [*sic*, instruction] means including a set of descriptions of message data;
>
> a virtual message processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the message instruction means that is arranged to provide directions for operation of the virtual message processor, whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task,
>
> wherein the *virtual machine means* is emulatable in different computers having incompatible hardwares or operating systems.

The specification discloses that a virtual machine can facilitate "portability" of programs:

> In conventional devices, each time a message is constructed or deconstructed, the operation of the machine will be handled by the application program. To change operation of the machine, the application must be changed. This is laborious, and gives rise to problems, as discussed above.
>
> The technique of creating a virtual processor (or in this case microprocessor) is well known and referred to as an interpreter. This allows programs to operate independent of processor. With the newer technique of also creating virtual peripherals then the whole is referred to as a "virtual machine".
>
> A virtual machine is computer programmed to emulate a hypothetical computer. Different incompatible computers may be programmed to emulate the same hypothetical computer. *Any computer programmed to emulate the hypothetical computer will thus be capable of executing programs for the virtual computer*. This creates a complete *portable* environment for program operations.

'945 Patent at 3:29-46 (emphasis added).

> The message processor means is preferably translated into the native code of the microprocessor in each hardware device on which the virtual machine is to be implemented. The message processor instructions are preferably virtual instructions to be expressed only in the language defined by the message processor means[—]and thus never requiring translation to any real hardware processor.

Id. at 4:5-11; *see id.* at 4:31-37 (similar as to "protocol processor means").

> In a preferred embodiment, therefore, a device in accordance with the present invention includes a virtual machine including virtual processors which are specifically arranged to control message construction, deconstruction, comparison and to control the communication of information, both for reception from a network and transmission to a network. These operations can therefore be carried

> out at speed, overcoming the problems with known virtual machines and interpreters, which tend to operate slower than conventionally programmed devices. The virtual machine therefore lends itself particularly to applications relating to communications, such as payment terminal devices and other devices in which message processing and communication comprise a significant proportion of the operation of the device. . . . The virtual machine can be implemented on any hardware, BIOS/OS arrangement and therefore facilitates *portability* of programs.

Id. at 4:51-5:3 (emphasis added).

During prosecution, Plaintiff similarly stated that using a virtual machine facilitates portability:

> As discussed in the [s]pecification . . . a virtual machine is a computer, which is programmed to emulate a hypothetical computer. This means that different incompatible computers (incompatible hardware and operating systems) may be programmed to emulate the same hypothetical computer. Applications may then be written for the hypothetical computer, which are therefore *portable* to the previously incompatible computers.

Dkt. No. 93, Ex. D, 11/18/2002 Response at 3 (emphasis added).

> One important feature of the Java language is that it can be interpreted by a Java Virtual Machine. Different versions of Java Virtual Machine are produced to interface with different underlying processors and operating systems. Thus, a program written in Java language may run on a variety of computers each having incompatible hardware or operating systems, and each running a Java Virtual Machine. Similar aspects of this type of a virtual machine has been described in the Specification . . . .

Id., Ex. N, 4/27/2004 Response at 12. Nowhere, however, did the patentee definitively state that all virtual machine applications must be portable or that a virtual machine can only run portable applications. *See Omega Eng'g v. Raytek Corp.*, 334 F.3d 1314, 1324 (Fed. Cir. 2003) ("As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public's reliance on *definitive* statements made during prosecution.") (emphasis added).

On balance, Defendants' proposal of "portable" is redundant and potentially confusing because portability of applications is merely a desired result of using a virtual machine. *See* '945 Patent at 3:29-46 ("Any computer programmed to emulate the hypothetical computer will thus be capable of executing programs for the virtual computer."). Nothing in the claims, the specification, or the prosecution history requires that all applications written for a virtual machine must be "portable." Indeed, the specification explains that in some embodiments, applications that can be executed on a virtual machine installed on a particular device might not operate, or at least not operate properly, when executed on a different device:

> Program Portability
>
> Portable Programs
>
> CardScript allows the writing of totally portable programs[;] it is also possible to write programs that are not very portable. Any CardScript program will "execute" on any CardScript enabled target, however *the result could be of no use on the target if special hardware characteristics are required for practical operation of the program*. CardScript provides a mechanism for avoiding the traps and keeping programs portable whilst still taking advantage of special hardware when available.

Id. at 22:21-31 (emphasis added).

In short, portability or non-portability of applications is not a limitation of the virtual machine that executes the applications. Defendants' proposal of "portable" is therefore hereby expressly rejected.

The Court accordingly hereby construes **"virtual machine means"** to mean **"a computer programmed to emulate a hypothetical computer for applications relating to transport of data."**

B. "virtual message processor"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"software implemented in the native code of the communications device that processes messages, including assembling, disassembling *and/or* comparing messages, for communication to and/or from a communications device"	"software implemented in the native code of the communications device that processes messages, including assembling, disassembling *and* comparing messages, for communication to and/or from a communications device"[4]

Dkt. No. 60, Ex. A at 3-4 (emphasis added). This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent.

At the November 20, 2013 hearing, Plaintiff agreed to adopt Defendants' proposed construction. The Court notes this agreement in Appendix A to this Claim Construction Memorandum and Order.

C. "virtual function processor"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"software which controls and/or selects general operations of a communication device"	"software separate from the operating system and drivers which controls and/or selects general operations of the communication device"

Dkt. No. 60, Ex. A at 5. This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "software which controls and/or selects general operations of a communication device."

[4] The parties' November 7, 2013 Joint Claim Construction Chart, filed pursuant to Local Patent Rule 4-5(d), revised Defendants' proposal so as to refer to "native code" rather than "the native code." Dkt. No. 98, Ex. A at 2. At the November 20, 2013 hearing, Defendants' presentation slides proposed "the native code." The Court therefore assumes that omission of the word "the" in the parties' Joint Claim Construction Chart was an error.

(1) The Parties' Positions

Plaintiff proposes the *CardSoft I* construction. *See CardSoft I* at 19-20. Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from re-litigating the construction of "virtual function processor." Dkt. No. 85 at 9. Alternatively, Plaintiff argues that "Defendants cannot . . . point to a single iota of intrinsic evidence, the claims, the common specification of the patents-in-suit or any paper from the prosecution of either patent-in-suit to support their proposed construction," specifically their proposal that the software must be "separate from the operating system and drivers." *Id.* at 10.

Defendants respond that Claim 1 of the '945 Patent, for example, recites a "virtual function processor" as something distinct from "operating systems." Dkt. No. 93 at 7. Defendants also argue that "[b]oth the 'hardware' and the 'operating system' are necessary elements of the claim, and both must be present, even if the incompatibility claimed pertains only to one or the other." *Id.* at 8-9. Defendants further explain that "the prosecution history makes clear that both [hardware and the operating system] must be present to satisfy the claim." *Id.* at 9.

Plaintiff replies that "Defendants conveniently ignore the word 'or' which appears between 'hardware' and 'operating systems' in the claim. Thus, contrary to Defendants' suggestion, it is not necessary that the 'virtual function processor' be separate and distinct from the operating system in the accused device for that device to satisfy all the limitations of the asserted claims." Dkt. No. 97 at 7. Plaintiff concludes that "so long as the accused virtual machine is capable of operating on computers having incompatible hardware, it is irrelevant whether those computers also have incompatible operating systems since the word 'or' makes those two options alternatives to one another." *Id.*

At the November 20, 2013 hearing, Defendants explained that their proposal of the phrase "separate from the operating system and drivers" means that the source code for the virtual function processor is a different "instruction set" than the source code for the operating system. For example, Defendants suggested, their proposed requirement of "separate" software would be satisfied if the virtual function processor software resided in different files than the operating system software. As to disclosures, in the specification, that the virtual function processor can control general operations of the communications device (*see, e.g.,* '945 Patent at 5:15-18), Defendants responded that the virtual function processor is disclosed as part of an *application* that drives the operation of the communications device by *using the underlying operating system*. Finally, Defendants emphasized that Figure 2 illustrates the virtual function processor as separate from the operating system, and Defendants argued that nothing in the specification discloses otherwise.

Plaintiff responded that the operating system is not part of the claimed virtual machine and is not a limitation of the claims. Plaintiff also argued that both an operating system and a virtual function processor could control operation of the device and could be written in the same native code.

Defendants replied that reading the virtual function processor as including the operating system would read the "wherein . . ." limitation (quoted below) out of the claims.

<u>(2) Analysis</u>

Claim 1 is representative and recites, in relevant part (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes
>
> a *virtual function processor* and function processor instructions for controlling operation of the device, and
>
> . . .

> a virtual message processor, which is arranged to be *called by the function processor* . . .
>
> wherein the virtual machine means is emulatable in different computers having *incompatible hardwares or operating systems*.

On one hand, the recital of "operating systems" in the claims suggests that the virtual function processor is distinct from the operating system because "we must presume that the use of . . . different terms in the claims connotes different meanings." *CAE Screenplates, Inc. v. Heinrich Fiedler GmbH & Co. KG,* 224 F.3d 1308, 1317 (Fed. Cir. 2000); *accord Primos, Inc. v. Hunter's Specialties, Inc.*, 451 F.3d 841, 848 (Fed. Cir. 2006) ("[T]he terms 'engaging' and 'sealing' are both expressly recited in the claim and therefore 'engaging' cannot mean the same thing as 'sealing'; if it did, one of the terms would be superfluous."); *Chi. Bd. Options Exch., Inc. v. Int'l Sec. Exch., LLC,* 677 F.3d 1361, 1369 (Fed. Cir. 2012) (noting "[t]he general presumption that different terms have different meanings"); *see Merck & Co. v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) ("A claim construction that gives meaning to all the terms of the claim is preferred over one that does not do so."); *see also Unique Concepts, Inc. v. Brown,* 939 F.2d 1558, 1562 (Fed. Cir. 1991) (noting that "[a]ll the limitations of a claim must be considered meaningful").

Likewise, the specification sometimes refers to the virtual machine as being distinct from the operating system of a device:

> The *virtual machine* can be implemented *on* any hardware, *BIOS/OS* [(operating system)] arrangement and therefore facilitates portability of programs.

'945 Patent at 5:1-3 (emphasis added). The patentee made a similar statement in the prosecution history:

> As discussed in the Specification . . . a virtual machine is a computer, which is programmed to emulate a hypothetical computer. This means that different incompatible computers (incompatible *hardware and operating systems*) may be programmed to emulate the same hypothetical computers.

Dkt. No. 93, Ex. D, 11/18/2002 Response at 3 (emphasis added). Finally, Figure 2 of the '945 Patent illustrates the function processor 107 in a different layer than the "BIOS/OS or hardware drivers." '945 Patent at 9:59-65 & 10:34-37.

On the other hand, the specification otherwise discloses the function processor as controlling "the device," in general terms:

> The virtual machine processors 103 also comprise a function processor 107 the operation of which is to *control and select general operations of the device* not specially controlled by the message and protocol processors 105, 106. *The function processor is also preferably implemented in the native code of the micro-processor of the hardware 100.* The application 104 includes protocol instructions 108, message instructions, 109, function support 110 and function instructions 111. The protocol instructions 106 govern operation of the protocol processor 106. The message instructions 109 provide directions for operation of the message processor 105. Function support 110 and function instructions 111 govern operation of the function processor 107. The application 104 and virtual machine 101, 102, 103 operate on data 112 input to the payment terminal to process it in accordance with the application 104.

Id. at 10:34-49 (emphasis added); *see id.* at 5:15-18 (similar).

On balance, nowhere did the patentee definitively state that a communication device must have an operating system or hardware drivers that are separate from the virtual function processor. *See Omega Eng'g*, 334 F.3d at 1324 ("As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public's reliance on *definitive* statements made during prosecution.") (emphasis added). Such a limitation should not be imported from preferred embodiments into the claims. *See Comark Commc'ns*, 156 F.3d at 1187.

Finally, Defendants have cited *Kustom Signals, Inc. v. Applied Concepts, Inc.*, 264 F.3d 1326, 1331 (Fed. Cir. 2001). *Kustom Signals*, however, dealt with whether the term "or" was exclusive, that is, whether it meant one or the other "but *not both*." *Id.* at 1330 (emphasis added). Defendants have not demonstrated that *Kustom Signals* warrants interpreting "or" to require that *both* of the elements recited in a disjunctive limitation must be present. *See* Dkt. No. 93 at 9.

In light of the foregoing, the Court hereby expressly rejects Defendants' proposal to include a "separate from the operating system and drivers" limitation in the construction. The Court accordingly hereby construes **"virtual function processor"** to mean **"software which controls and/or selects general operations of a communication device."**

D. "function processor instructions"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"a set of instructions that control operation of the communications device"	"a set of application program instructions that control operation of the communications device"

Dkt. No. 60, Ex. A at 7. This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "a set of instructions that control operation of the communications device."

<u>(1) The Parties' Positions</u>

Plaintiff proposes the *CardSoft I* construction. *See CardSoft I* at 24-25. Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from re-litigating the construction of "function processor instructions." Dkt. No. 85 at 10. Alternatively, Plaintiff argues that "Defendants' can point to no support for their proposition that the function processor instructions must be part of an "application program" (as opposed to, for example, some other software loaded onto the communications device)." *Id.*

at 11. Plaintiff further argues that Defendants' proposal contradicts the plain language of Claim 8 of the '945 Patent. *Id.*

Defendants respond that the specification describes "function instructions" as being part of an application. Dkt. No. 93 at 20 (citing '945 Patent at 5:26-30, 10:40-47 & 12:35-42). Defendants submit that "Plaintiff does not point to any support in the intrinsic record tending to indicate that the 'function processor instructions' can be anything other than 'application program instructions.'" *Id.* at 20.

At the November 20, 2013 hearing, Defendants urged that without their proposed "application program" instructions limitation, the construction would be overbroad because the specification consistently points to the application as the source of the instructions. Plaintiff responded that such disclosures in the specification explicitly refer to preferred embodiments.

(2) Analysis

Claim 1 recites, in relevant part (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes
>
> a virtual function processor and *function processor instructions for controlling operation of the device*, and
>
> . . .
>
> a virtual message processor, which is arranged to be called by the function processor . . . .

Plaintiff has also submitted that Claim 8 provides useful context:

> 8. A device in accordance with claim 1, wherein the device includes a microprocessor which runs in accordance with native software code, and wherein the function processor instruction means are implemented in software defined by the function processor means and do not require translation to the native code of the microprocessor.

The specification discloses that the "virtual machine" or the "application" can include function processor instructions:

> *The virtual machine* preferably also includes a function processor means arranged to control overall virtual machine action in response to operator or other external events, and *also preferably includes function processor instructions* which are arranged to *provide directions for operation of the function processor means*.
>
> * * *
>
> *In the preferred embodiment, the "application" will therefore comprise instructions for the* message, protocol and *function processor means*. The instructions for the function processor means may include such prior art modules as a function event schedular [*sic*] and function selector.
>
> * * *
>
> *The application 104 includes* protocol instructions 108, message instructions[] 109, function support 110 and *function instructions 111*. . . . Function support 110 and *function instructions 111 govern operation of the function processor 107*.

'945 Patent at 5:9-14, 5:26-30 & 10:40-47 (emphasis added); *see id.* at 12:43-52.

On balance, function processor instructions are not defined in any of the intrinsic evidence as necessarily being part of an "application program." Instead, such a limitation is disclosed as part of a preferred embodiment and should not be imported into the claims. *See Comark Commc'ns*, 156 F.3d at 1187. Defendants' proposal of that language is therefore hereby expressly rejected.

The Court accordingly hereby construes **"function processor instructions"** to mean **"a set of instructions that control operation of the communications device."**

E. "emulatable in different computers having incompatible hardwares or operating systems"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"capable of executing programs on different computers having incompatible hardware or operating systems"	"Should be construed separately as 'capable of executing the same application program instructions on different computers having incompatible hardware or operating systems,' and 'the different communication devices have different hardware or operating systems such that absent the virtual machine the same application program instructions cannot be executed without modification.'"[5]

Dkt. No. 60, Ex. A at 8; Dkt. No. 93 at 16; *see* Dkt. No. 98, Ex. A at 7-8. This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "capable of executing programs on different computers having incompatible hardware or operating systems."

(1) The Parties' Positions

Plaintiff proposes the *CardSoft I* construction. *See CardSoft I* at 14-17. Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from re-litigating the construction of "emulatable in different computers having incompatible hardwares or operating systems." Dkt. No. 85 at 12. Alternatively, Plaintiff argues that "Defendants again seek to improperly impose limitations on the scope of the claims that are not supported by any of the intrinsic evidence in this case, in this instance by separating this limitation into two and then imposing stark restrictions on both . . . ." *Id.* Plaintiff further argues:

[5] Defendants originally proposed: "The term is construed separately as set forth below (e.g., 'the virtual machine means is emulatable' and 'in different computers having incompatible hardwares or operating systems')." Dkt. No. 60, Ex. A at 8.

> [T]here is nothing in the claims, the common specification or the respective prosecution file histories that requires that the virtual machine be capable of executing the same "application program instructions" on different computers having incompatible hardware or operating systems. Rather, as noted by this Court when construing this limitation in *CardSoft I*, the specification only requires that the virtual machine be capable of executing programs – not application program instructions – on different computers.

Id. (citing *CardSoft I* at 16-17); Dkt. No. 85 at 19 ("[T]here is nothing in the language of the claims, or in the common specification, that requires the virtual machine to be capable of running the same unmodified application on multiple different communication devices.").

Defendants respond:

> To address the portability problem, the inventor claimed a specialized virtual machine that can be used as the target for application programs regardless of the underlying software and hardware components of the device. That is, the virtual machine, which is emulated in different devices, allows the same application program to be executed on different devices since the application programs are targeted for and executed by the same virtual machine which functions as a common interface for the application program.

Dkt. No. 93 at 12. Defendants also note that whereas Plaintiff's opening brief touts the benefit of enabling applications to run on multiple types of devices without re-certification of the applications, "the only reason that an application would not require re-certification is that it is the same application that has already been certified." *Id.* at 13.

Plaintiff replies that *CardSoft I* expressly rejected an attempt to exclude, from the scope of claims, embodiments in which source code is compiled differently for different target devices. Dkt. No. 97 at 8 (citing *CardSoft I* at 12).

At the November 20, 2013 hearing, Defendants stressed that "incompatible" means more than merely "different." Defendants also emphasized disclosure in the specification distinguishing prior art in which programs had to be modified in order to operate on different devices. *See* ‘945 Patent at 3:5-12. Defendants reiterated that the purpose, and necessary effect, of using a virtual machine is that no modifications are necessary.

Plaintiff responded that the disclosures cited by Defendants pertain to re-writing an entire application rather than to merely recompiling the same source code for different devices.

(2) Analysis

Defendants’ proposal for the term “the virtual machine means is emulatable” would require that what can be executed are “the same application program instructions.” Dkt. No. 60, Ex. A at 12. Similarly, Defendants’ proposal for the term “in different computers having incompatible hardwares or operating systems” would require that “the different communication devices have different hardware or operating systems such that absent the virtual machine the *same application program instructions* cannot be executed *without modification*.” *Id.* at 14-15 (emphasis added). Both of these proposals center on whether the virtual machine must enable running the same application program instructions on different devices. With this context in mind, the Court turns to the intrinsic evidence.

Claim 1 of the ‘945 Patent is representative and recites:

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means . . .
> . . .
> wherein the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems.

The specification states that prior art systems suffered from the disadvantage of programs not being “portable”:

> Prior art payment terminal devices are generally programmed in a conventional manner. That is, programming comprises a sequential set of operating instructions which are executed in sequence to carry out a remote payment transaction. This “sequential program” may be directly compiled onto the processor of the device so that the device is under direct program control or, as is more usual, an applications program in a conventional programming language

> may control operations through a BIOS/OS. Whatever conventional programming form is used, however, the device suffers from the problems which are discussed in the preamble of this specification. The programs are *not portable* between devices having different hardware or operating system architectures and it is necessary to write a program specifically for each type of device. Further, any amendments to the operation of the device must be programmed by a programmer having knowledge of that particular device and program arrangement.

‘945 Patent at 9:37-54 (emphasis added)

> [P]rogramming will be different for different types of devices, which may have different hardware arrangements as well, and must be carried out separately for each different type of device (i.e., different reprogramming operations must be carried out for different devices even where the same operational alterations may be required).
>
> *The programming alterations are not “portable” between different types of devices*.

Id. at 3:7-14 (emphasis added). In light of this, the specification discloses, emulating a virtual computer can be advantageous:

> A virtual machine is computer programmed to emulate a hypothetical computer. Different *incompatible computers may be programmed to emulate the same hypothetical computer*. Any computer programmed to emulate the hypothetical computer will thus be capable of executing programs for the virtual computer. This creates a complete portable environment for program operations.

‘945 Patent at 3:40-46 (emphasis added); *see id.* at 5:7-8 (“Each brand [of payment terminal device] is seen by the application as the same virtual machine.”).

During prosecution, the patentee similarly stated:

> As discussed in the Specification . . ., a virtual machine is a computer, which is programmed to emulate a hypothetical computer. This means that different incompatible computers (incompatible hardware and operating systems) may be programmed to emulate the same hypothetical computer. Applications may then be written for the hypothetical computer, which are therefore portable to the previously incompatible computers.

Dkt. No. 93, Ex. D, 11/18/2002 Response at 3-4; *see id.*, Ex. I, 10/14/2004 Amendment at 12-14 (similar).

As to extrinsic evidence, Defendants have cited two technical dictionary definitions of "emulation." The *Dictionary of Computer and Internet Terms* defines "emulation" as "the process of achieving the same results as if you had a different machine than the one you're actually using." Dkt. No. 93 at Ex. J. The *IEEE Standard Dictionary of Electrical and Electronics Terms* defines "emulation" as "[a] model that accepts the same inputs and produces the same outputs as a given system." *Id.* at Ex. K. Defendants have also cited a general-purpose dictionary definition, in *Webster's Unabridged Dictionary*, that defines "emulate" in the computer context as "to imitate (a particular computer system) by using a software system, often including a microprogram or another computer that enables it to do the same work, run the same programs, etc., as the first." *Id.* at Ex. L. These definitions are consistent with the above-quoted usage of "emulate" in the specification. *See* '945 Patent at 3:40-46; *see also id.* at 24:36-40 (PC Simulator / There is a[n] implementation of the CardScript Virtual machine available on the PC that not only runs your program, it also *emulates* the keyboard layout and other controls of any target machine.") (emphasis added).

On balance, Defendants' proposal to refer to "the same application program instructions" is redundant and potentially confusing because portability of applications is merely a desired result of using a virtual machine. *See id.* at 3:29-46 ("Any computer programmed to emulate the hypothetical computer will thus be capable of executing programs for the virtual computer."). Nothing in the claims, the specification, or the prosecution history requires that the ability to emulate a particular virtual machine on different devices requires that the virtual machine can properly execute the same application program instructions on different devices.

For example, although the disputed term requires that the virtual machine can be emulated on different devices, the specific capabilities of the devices or the virtual machine are not limitations. To the contrary, the specification explains that in some embodiments, applications that operate on a virtual machine installed on a particular device might not operate, or at least not operate properly, on a different device:

> Program Portability
>
> Portable Programs
>
> CardScript allows the writing of totally portable programs[;] it is also possible to write programs that are not very portable. Any CardScript program will "execute" on any CardScript enabled target, however *the result could be of no use on the target if special hardware characteristics are required for practical operation of the program*. CardScript provides a mechanism for avoiding the traps and keeping programs portable whilst still taking advantage of special hardware when available.

Id. at 22:21-31 (emphasis added).

In short, the Court has construed the term "virtual machine means," above, and the present disputed term addresses the devices on which the virtual machine can be emulated rather than any specific program execution capabilities.

Finally, as noted above, Plaintiff interprets *CardSoft I* as having rejected an argument that the "emulatable" limitation "preclude[s] the source code from being compiled for different target computers." Dkt. No. 97 at 8. Yet, the *CardSoft I* statements cited by Plaintiff pertain to the source code for the virtual machine, *not* the source code for the application that the virtual machine is executing:

> [C]olumn 3, lines 29-55 of the specification . . . criticizes prior art virtual machines for requiring applications written in hardware-specific code since such applications would not be portable to different devices. '945 Patent at 3:37-54. It does not, however, discuss *whether the virtual machine itself can be written in hardware-specific code* – indeed, the cited portion is silent on the topic of the *code used to implement the claimed virtual machine*. Likewise, *none of the other specification language to which Defendants cite states that the virtual machine, or any part thereof, must necessarily be written in a hardware/operating system independent language* in order to be emulatable in different computers.

CardSoft I at 12 (emphasis added); *see* Dkt. No. 93, Ex. N, 4/27/2004 Response at 12 (”Different versions of Java Virtual Machine are produced to interface with different underlying processors and operating systems.”). Thus, in reaching its present findings regarding the “emulatable” limitation, the Court does not adopt or rely upon Plaintiff’s interpretation of the above-quoted portion of *CardSoft I*. Moreover, based on the arguments presented in the parties’ briefing and at the November 20, 2013 hearing, any underlying dispute about the recompiling of source code is a factual dispute regarding infringement rather than a legal dispute regarding claim construction. *See PPG Indus. v. Guardian Indus. Corp.*, 156 F.3d 1351, 1355 (Fed. Cir. 1998) (noting that “the task of determining whether the construed claim reads on the accused product is for the finder of fact”).

The Court therefore hereby construes **“emulatable in different computers having incompatible hardwares or operating systems”** to mean **“capable of executing programs on different computers having incompatible hardware or operating systems.”**

F. “called by the function processor”

Plaintiff’s Proposed Construction	**Defendants’ Proposed Construction**
“The ‘function processor’ has the construction stated above with respect to ‘virtual function processor.’ The remaining terms have their plain and ordinary meaning and do not require further construction.”	“the virtual function processor provides the message instruction to the message processor when a message is to be processed”

Dkt. No. 60, Ex. A at 8. This term appears in Claims 1 and 14 of the ‘945 Patent and Claim 1 of the ‘683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: “Plain and ordinary meaning.”

(1) The Parties' Positions

Plaintiff argues that because Defendants are successors-in-interest to Hypercom, "[p]ermitting Defendants to now raise additional terms as allegedly requiring construction would be to provide a road map for future accused infringers seeking to avoid the effect of an adverse claim construction ruling—just form a new entity and then raise additional claim terms for construction in a subsequent action." Dkt. No. 85 at 13. Alternatively, Plaintiff argues that Defendants' proposed construction would improperly limit the disputed term to a preferred embodiment disclosed in the specification. *Id.* at 14.

Defendants respond that "collateral estoppel cannot bar the litigation of the proper construction of claim terms that were not previously construed in *CardSoft I*." Dkt. No. 93 at 4. Defendants argue that the specification and the prosecution history explain that all messages are handled by the virtual message processor. *Id.* at 23.

At the November 20, 2013 hearing, Plaintiff stated that it had no objection to the Court's preliminary construction. Defendants stated that they, too, had no objection, provided that the Court also adopts its preliminary construction for the term "when a message is required to be handled by the communications device, the message processor is called to carry out the message handling task," which is a disputed term addressed separately below.

(2) Analysis

CardSoft I did not address this disputed term. Claim 1 recites, in relevant part (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes
>
> a virtual function processor and function processor instructions for controlling operation of the device, and
>
> . . .

> a virtual message processor, which is arranged to be *called by the function processor* and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the message instruction means that is arranged to provide directions for operation of the virtual message processor, whereby *when a message is required to be handled by the communications device the message processor is called to carry out the message handling task*,
>
> . . . .

The specification discloses:

> The *function processor 107 fetches the message instruction 109* for this event and *the message processor 105 then operates* to load the data from the card into labelled [*sic*] fields (steps 5, 6 and 7) *according to the message instructions*.

See id. at 12:63-67 (emphasis added).

Figure 3 similarly discloses that the "message processor places data from card into fields as directed by message instruction means," although Figure 3 is described in the specification as being merely "an *example* of an operation of the device, for one typical step in a remote payment transaction." *Id.* at 11:45-46 (emphasis added); *see MBO Labs., Inc. v. Becton, Dickinson & Co.*, 474 F.3d 1323, 1333 (Fed. Cir. 2007) (noting that "patent coverage is not necessarily limited to inventions that look like the ones in the figures").

Finally, during prosecution the patentee discussed calling the message processor to handle messages:

> By a single call, *the virtual function processor can call the virtual message processor* ([*see* '945 Patent at 12:53-67], where the SAVE instruction calls the message processor) *in order to deal with the message*. If message handling was carried out by the function processor (conventional virtual machine), the operation would be slower. It would be necessary for separate instructions for the message handling process to be provided to the conventional virtual processor. *By specifically arranging a message processor* (being a series of subroutines purely for handling message[s]), *the operation runs faster*.

Dkt. No. 93, Ex. D, 11/18/2002 Response at 4 (emphasis added).

> [T]he communication device as described and presently claimed is quite significantly different from the Java Virtual Machine of *Stern* [(United States

> Patent No. 5,935,249)], because the presently claimed invention includes a *dedicated virtual message processor*, which function is to perform generic handling of messages.

Id., Ex. N, 4/27/2004 Response at 12 (emphasis added); *see id.* at 11-14.

Nonetheless, Defendants have not shown what their proposed construction would substantively add to the plain language of the claim. At the November 20, 2013 hearing, Defendants agreed that the antecedent basis for "the function processor" is "virtual function processor." The Court has already construed the term "virtual function processor," above, so the phrase "the function processor" does not require additional construction here. The remaining phrase—"called by"—requires no construction, particularly because other claim language recites that "when a message is required to be handled by the communications device[,] the message processor is called to carry out the message handling task." '945 Patent at Claims 1 & 14; '683 Patent at Claim 1. That language is also a disputed term in the above-captioned case and is addressed separately, below.

The Court therefore hereby construes **"called by the function processor"** to have its **plain and ordinary meaning**.

G. "assembling the messages, disassembling messages and comparing the messages"

Plaintiff's Proposed Construction	Defs.' Proposal
Plain and ordinary meaning, does not require further construction beyond that identified above. To the extent that the Court does determine that construction of the terms "assembling the messages" or "disassembling messages" beyond their ordinary meaning is necessary, then [Plaintiff] proposes for "assembling the messages": "to take data and build a message in accordance with the message description and place the built message in a location" Likewise, [Plaintiff] alternatively proposes for "disassembling messages": "to take a message and deconstruct it into various components and place the various components into other locations"	"loading and extracting data from fields of messages and comparing data from fields of the messages"

Dkt. No. 60, Ex. A at 9. This term appears in Claim 1 of the '945 Patent and Claim 1 of the '683 Patent.

The parties' November 7, 2013 Joint Claim Construction Chart, filed pursuant to Local Patent Rule 4-5(d), states that the parties have reached agreement that this disputed term should be given its "[p]lain and ordinary meaning, does not require further construction." Dkt. No. 98, Ex. A at 9-10. The Court notes this agreement in Appendix A to this Claim Construction Memorandum and Order.

H. "under the direction of the message instruction means"

Plaintiff's Proposed Construction	Defendants' Proposed Construction
"The 'message instruction means' has the construction stated above (and agreed to by the parties). The remaining terms have their plain and ordinary meaning and do not require further construction."	"as defined by the descriptions of the fields of the message provided by the message instructions"

Dkt. No. 60, Ex. A at 10; Dkt. No. 85 at 15; *see* Dkt. No. 98, Ex. A at 10-11. This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "Plain and ordinary meaning."

(1) The Parties' Positions

Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from raising this term as one requiring construction. Dkt. No. 85 at 15. Alternatively, Plaintiff argues that "there is nothing in the common specification of the patents-in-suit that requires the message instruction means to direct the tasks of the virtual message processor using only the descriptions of the fields of the message." *Id.* at 16.

Defendants respond:

> [A]s can be seen from the claim element [in which the disputed term appears], the message instruction means directs the virtual message processor to carry out the message handing tasks. . . . It is clear from the specification that the message instruction means provides directions via the content of message fields. . . . The specification does not disclose any other way for the message instructions means to direct the virtual message processor.

Dkt. No. 93 at 21 (citing '945 Patent at 13:36-38, 14:3-16 & 40:55-64).

At the November 20, 2013 hearing, Defendants argued that other claim language provides support for their proposed construction, in particular the recital of "a set of descriptions of message data." Plaintiff responded that the claims recite a "message in[str]uction means including a set of descriptions of message data," and Plaintiff noted that "including" is a non-limiting word. *See* '945 Patent at Claim 1.

(2) Analysis

CardSoft I did not address this disputed term. As noted in Appendix A to this Claim Construction Memorandum and Order, however, the parties have agreed upon a construction for the constituent term "message instruction means."

Claim 1 of the ‘945 Patent is representative and recites, in relevant part (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes
>
> . . .
>
> *message in*[*str*]*uction means including a set of descriptions of message data;*
>
> a virtual message processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages *under the direction of the message instruction means* that is arranged to provide directions for operation of the virtual message processor, whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task,
>
> . . . .

The specification discloses that a message can include “fields” or “components,” but the specification also suggests that a message as a whole can be compared:

> Each message instruction will therefore include a description of a *field of message data*, providing instruction for the virtual message processor means which enable it to carry out a number of tasks:
>
> 1. *To compare a message with a message description* to see if it is the correct required message.
>
> 2. To take a message of the correct description from a location and place it in another location.
>
> 3. To take a message and deconstruct it into various *components* and place the various components into other locations.
>
> 4. To take data and build a message in accordance with the message description and place the built message in a location.
>
> 5. *Compare one message with another message*.

‘945 Patent at 14:7-17 (emphasis added).

> Messages
>
> Output messages
> A messages buffer is built from *fields* in the data dictionary, and from strings in much the same way a printout is build [*sic*, built]. . . .
>
> Input Messages
> Messages are also used to specify how data is transferred from a received buffer and stored in data *fields*. . . .
>
> The Message Engine (Processor)
> The message engine is used both to transfer information both from data *fields* not [*sic*, into] a message buffer, and from a message buffer to data *fields*[.]

Id. at 37:35-50 (emphasis added).

The specification also discloses message "fields" that are converted into message instructions:

> *Each message is made up of a number of message "fields" 120.* In this particular example, there are seven fields . . . . *Each of the seven field[s] is converted to a message instruction* for use by the virtual message processor. This is the information which is typically found on any magnetic stripe card. The message instructions in accordance with this embodiment direct the message processor to process these elements. Each field is associated with descriptors which provide further instructions for the handling of that element.

'945 Patent at 14:56-67 (emphasis added).

On balance, these above-quoted passages do not compel a finding that a "message" is merely a collection of fields or that message instructions must pertain to fields. Instead, for example, the specification suggests that "comparing messages" includes comparing one message to another or to a message description. *Id.* at 14:7-17 (quoted above). Further, even if the specification were interpreted as requiring fields, such a limitation is a feature of preferred embodiments that should not be imported into the comparatively generic claim language, which recites "messages." *See Comark Commc'ns*, 156 F.3d at 1187.

As for the prosecution history, the patentee distinguished the "Stern" reference, United States Patent No. 5,935,249, which the patentee characterized as "describ[ing] merely a JavaOS being operable on different processors supporting the Java Virtual Machine." Dkt. No. 93, Ex. I, 10/14/2004 Amendment at 9. The patentee argued that Stern "provides absolutely no teaching of providing descriptions of message data to control processing of messages." *Id.* at 10-11. For example, the patentee stated:

> [T]he message instruction means of the presently claimed invention includes a set of descriptions of message data. The descriptor 121 designates the characteristic to be applied to the information in a particular field of a message. This approach to processing messages is not at all seen in *Stern*.

Id. at 10; *see* '945 Patent at 13:36-38 (discussing "descriptors 121"). On balance, however, Defendants have not identified any definitive statements that would warrant requiring "descriptions of the fields of the message" as a claim limitation. *See Omega Eng'g*, 334 F.3d at 1324 ("As a basic principle of claim interpretation, prosecution disclaimer promotes the public notice function of the intrinsic evidence and protects the public's reliance on *definitive* statements made during prosecution.") (emphasis added). Instead, for example, the patentee referred to the "descriptor 121" of a particular preferred embodiment but then referred more general to "[t]his approach." Dkt. No. 93, Ex. I, 10/14/2004 Amendment at 10.

The Court therefore hereby expressly rejects Defendants' proposed construction. No further construction is required. *U.S. Surgical Corp. v. Ethicon, Inc.*, 103 F.3d 1554, 1568 (Fed. Cir. 1997) ("Claim construction is a matter of resolution of disputed meanings and technical scope, to clarify and when necessary to explain what the patentee covered by the claims, for use in the determination of infringement. It is not an obligatory exercise in redundancy."); *see also O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008) ("[D]istrict courts are not (and should not be) required to construe every limitation present in a patent's asserted claims."); *Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1207 (Fed. Cir. 2010) ("Unlike *O2 Micro*, where the court failed to resolve the parties' quarrel, the district court rejected Defendants' construction.").

The Court accordingly hereby construes **"under the direction of the message instruction means"** to have its **plain and ordinary meaning**.

I. "when a message is required to be handled by the communications device, the message processor is called to carry out the message handling task"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"The term 'message processor' has the construction stated above with respect to 'virtual message processor.' The remaining terms have their plain and ordinary meaning and do not require further construction."	"all messages are handled by the virtual message processor"

Dkt. No. 60, Ex. A at 11; *see* Dkt. No. 98, Ex. A at 12-13. This term appears in Claims 1, 12, and 14 of the '945 Patent and Claim 1 of the '683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: "all messages handled by the communications device are handled by the virtual message processor."

(1) The Parties' Positions

Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from raising this term as one requiring construction. Dkt. No. 85 at 16. Alternatively, Plaintiff argues that "there is nothing in the claims or the common specification of the patents-in-suit that requires the virtual message processor to handle all of the message[s] that the communications device may send or receive." *Id.* at 17. Plaintiff also argues that Defendants' proposal is inconsistent with disclosure in the specification of (in

Plaintiff's words) "additional software known as a protocol processor." *Id.* (citing '945 Patent at 15:49-51 & 15:67-16:1).

Defendants respond that the specification and the prosecution history state that all messages are handled by the virtual message processor. Dkt. No. 93 at 23 (citing '945 Patent at 12:63-67 & Fig. 3; citing Dkt. No. 93, Ex. I, 10/14/2004 Amendment at 4; citing *id.*, Ex. N, 4/27/2004 Response at 10-13).

At the November 20, 2013 hearing, Plaintiff had no objection to the Court's preliminary construction except as to the word "all." Defendants responded that the virtual message processor definitely has to touch all messages.

(2) Analysis

CardSoft I did not address this disputed term. Claim 1 recites, in relevant part (emphasis added):

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means which includes
>
> a virtual function processor and function processor instructions for controlling operation of the device, and
>
> . . .
>
> a virtual message processor, which is arranged to be called by the function processor and which is arranged to carry out the message handling tasks of assembling the messages, disassembling messages and comparing the messages under the direction of the message instruction means that is arranged to provide directions for operation of the virtual message processor, whereby *when a message is required to be handled by the communications device the message processor is called to carry out the message handling task*,
>
> . . . .

The specification discloses a message processor performing message construction, deconstruction, and comparison:

> The message processor 105 is a series of several sub-routines implemented in the native code of the CPU 1, the specific operation of which is to *construct, deconstruct and compare messages* in accordance with message instructions 109 from the application 104. . . . The function processor 107 fetches the message

> instruction 109 for this event and the message processor 105 then operates to *load the data from the card into labelled fields* (steps 5, 6 and 7) according to the message instructions.

Id. at 12:56-67 (emphasis added).

> *The message processor is responsive to all the message instructions* to load the data from the magnetic stripe card into the appropriate fields with the appropriate formats in accordance with all the rules designated in the instructions.
>
> This embodiment of the present invention includes another class of message instruction means, known as a "Form". Instead of a Data Representation as a message descriptor, a Form includes description of a Location of the data field in the Form. FIG. 8 is a display provided by a development tool enabling the programmer to prepare message instructions for a Form message. On the left hand side of the display a panel 70 illustrates Form layout. The fields in the Form include MerName, Address Line 1, etc. The location of these fields can be moved within the panel 70. The location in the panel is provided as a descriptor and for the message instruction. The Form type of message instruction controls displays, reports, print-outs, and the like. The type of Form is given by the instruction designated by reference numeral 71, in the example illustrated in FIG. 8 being a print-out. *The message processor takes the fields from known memory locations or other locations and enters them in the locations enabling the Form described by the Form instruction to be produced.*

'945 Patent at 15:19-41 (emphasis added).

Likewise, during prosecution the patentee repeatedly referred to handling of messages by the virtual message processor:

> [A] separate virtual processor, the *virtual message processor*, is provided, *the specific function of which is to disassemble, assemble, and compare messages.*
>
> * * *
>
> *If message handling was carried out by the function processor (conventional virtual machine), the operation would be slower.* It would be necessary for separate instructions for the message handling process to be provided to the conventional virtual processor. By *specifically arranging a message processor* (being a series of subroutines purely for handling message), the operation runs faster.

Dkt. No. 93, Ex. D, 11/18/2002 Response at 3 & 4 (emphasis added).

> [N]owhere in Stern (or in any other cited references) discloses *the claimed virtual message processor that carries out the presently claimed task of --assembling, disassembling and comparing messages, whereby when a message is required to be handled by the communications dev[i]ce the message processor is called to carry out the message handling task*-- . . . .
>
> * * *
>
> [T]he communication device as described and presently claimed is quite significantly different from the Java Virtual Machine of Stern, because the presently claimed invention includes a *dedicated virtual message processor*, which function is to perform generic handling of messages.
>
> Applicant respectfully submits that Java Virtual Machine does not include such *claimed dedicated virtual message processor*. The addition of such a claimed message processor represents an additional component that must be added to any known Java Virtual Machine. This introduction of the *dedicated virtual message processor in the presently claimed invention* provides quite substantially different handling of the messages than in any known Java Virtual Machine. In other words, the Java language takes a different approach than the functions of the *presently claimed invention that includes a dedicated message processor*.
>
> * * *
>
> Stern fails to teach the *claimed dedicated "virtual" message processor* when, for example, called by the "virtual" function processor --*carry out the task of assembling, disassembling and comparing messages, whereby when a message is required to be handled by the communications device the message processor is called to carry out the message handling task*--.
>
> As stated in the Specification . . ., providing a *separate virtual message processor* allows for "faster, simpler programming." Neither Stern nor Yan [(United States Patent No. 6,003,065)] teaches or suggests the provision of the *claimed virtual machine with a dedicated virtual message processor*. That is, if a Java Virtual Machine as described in Stern is used to perform messaging, each application developed would be required to adjust to the characteristics of the different devices that the application was to execute on, such as screen width and fonts.
>
> The claimed virtual message processor removes this burden from the development of the application and places it on the software platform that resides on the device. This relieves the application developers of the burden of programming to the physical characteristics of the platform that [the] application will execute on.

Id., Ex. N, 4/27/2004 Response at 11-14 (emphasis modified); *see id.*, Ex. I, 10/14/2004 Amendment at 9-10, 11-12 & 13 (similar).

> [T]he novelty lies in that the messages are compared under the direction of the messages [*sic*] instructions of the virtual message processor which is a part of the virtual machine means.

Id. at 9-10.

The patentee thus repeatedly explained that the message processor is called upon for message handling, and no other structure is disclosed as handling messages in the manner recited by the claims. The disputed term should be construed in light of this context. *See Retractable Techs., Inc. v. Becton, Dickinson & Co.*, 653 F.3d 1296, 1305 (Fed. Cir. 2011) ("In reviewing the intrinsic record to construe the claims, we strive to capture the scope of the actual invention, rather than strictly limit the scope of claims to disclosed embodiments or allow the claim language to become divorced from what the specification conveys is the invention."); *see also Nystrom v. TREX Co., Inc.*, 424 F.3d 1136, 1144-45 (Fed. Cir. 2005) (construing term "board" to mean "wood cut from a log" in light of the patentee's consistent usage of the term; noting that patentee "is not entitled to a claim construction divorced from the context of the written description and prosecution history"); *Am. Piledriving Equip., Inc. v. Geoquip, Inc.*, 637 F.3d 1324, 1333 (Fed. Cir. 2011) ("[T]he consistent reference throughout the specification to the 'eccentric weight portion' as structure extending from the face of the gear makes it apparent that it relates to the invention as a whole, not just the preferred embodiment.").

On balance, these numerous, consistent statements in the specification and the prosecution history warrant construing the disputed term to require that all messages handled by the communications device are handled by the virtual message processor. *See Typhoon Touch Techs., Inc. v. Dell, Inc.*, 659 F.3d 1376, 1381 (Fed. Cir. 2011) ("The patentee is bound by representations made and actions that were taken in order to obtain the patent."); *see generally Computer Docking Station Corp. v. Dell Inc.*, 519 F.3d 1366, 1374-75 (Fed. Cir. 2008).

Plaintiff has argued that "if the protocol processor is responsible for controlling and organizing the flow of messages to and from of [*sic*] the communications device[], then the virtual message processor is not necessarily handling all of the messages." Dkt. No. 85 at 17. The specification discloses: "The protocol processor 106 is arranged to organise [*sic*] communications, in accordance with directions from the protocol processor instructions 108. * * * Protocol instructions describe message flow both from and to the device." '945 Patent at 15:49-51 & 15:67-16:1. A "protocol processor" is similarly recited in several dependent claims as being "arranged to organize communications to and from the device." *See id.* at Claims 2, 13 & 16. Nonetheless, because the protocol processor is for organizing communications to and from the device rather than for assembling, disassembling, and comparing messages, the evidence cited by Plaintiff regarding the protocol processor is complementary of, rather than inconsistent with, construing the disputed term to required that all messages are handled by the virtual message processor.

In light of these disclosures cited by Plaintiff, however, the Court hereby clarifies that the messages handled by the communications device need not be handled *solely* by the virtual message processor. In other words, other components (such as the protocol processor) may have a role in handling a particular message so long as the virtual message processor also handles the message.

The Court therefore hereby construes **"when a message is required to be handled by the communications device, the message processor is called to carry out the message handling task"** to mean **"all messages handled by the communications device are handled by the virtual message processor."**

J. “the virtual machine means is emulatable”

Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“The ‘virtual machine’ has the construction stated above with respect to ‘virtual machine means;’ the remaining terms have their plain and ordinary meaning and do not require further construction”[6]	“capable of executing the same application program instructions on different computers having incompatible hardware or operating systems”

Dkt. No. 60, Ex. A at 12; Dkt. No. 98, Ex. A at 13-14. This term appears in Claims 1, 12, and 14 of the ‘945 Patent and Claim 1 of the ‘683 Patent.

The Court has effectively construed this disputed term as part of the term “emulatable in different computers having incompatible hardwares or operating systems,” addressed above. As a result, no separate construction is necessary for “the virtual machine means is emulatable.”

K. “in different computers having incompatible hardwares or operating systems”

Plaintiff’s Proposed Construction	Defendants’ Proposed Construction
“This is part of the construction stated above e.g. ‘emulatable in different computers having incompatible hardwares or operating systems;’ does not require separate construction”[7]	“the different communication devices have different hardwares or operating systems such that absent the virtual machine the same application program instructions cannot be executed without modification”[8]

[6] Plaintiff originally proposed: “The term ‘virtual machine’ has the construction stated above with respect to ‘virtual machine means.’ The term “emulatable’ is part of the claim construction stated above with respect to ‘emulatable in different computers having incompatible hardwares or operating systems[]’ and does not require separate construction.” Dkt. No. 60, Ex. A at 12. Plaintiff proposed in its briefing: “This is only part of a claim limitation and should not be construed in isolation. Rather, this Court’s prior construction of the ‘emulatable’ limitation should be adopted.” Dkt. No. 85 at 18.

[7] Plaintiff originally proposed: “This is part of the construction stated above with respect to ‘emulatable in different computers having incompatible hardwares or operating systems[]’ and does not require separate construction.” Dkt. No. 60, Ex. A at 14-15. Plaintiff proposed in its briefing: “This is only part of a claim limitation and should not be construed in isolation. Rather, this Court’s prior construction of the ‘emulatable’ limitation should be adopted.” Dkt. No. 85 at 19.

[8] In their briefing, Defendants proposed “hardware” instead of “hardwares.” *Compare* Dkt. No. 93 at 14 *with* Dkt. No. 60, Ex. A at 14-15 & Dkt. No. 98, Ex. A at 14-15.

Dkt. No. 60, Ex. A at 14-15; Dkt. No. 98, Ex. A at 14-15. This term appears in Claims 1, 12, and 14 of the ‘945 Patent and Claim 1 of the ‘683 Patent. Shortly before the start of the November 20, 2013 hearing, the Court provided the parties with the following preliminary construction: “Plain and ordinary meaning.”

CardSoft I construed “emulatable in different computers having incompatible hardwares or operating systems” to mean “capable of executing programs on different computers having incompatible hardware or operating systems.” *See CardSoft I* at 14-17.

(1) The Parties’ Positions

Plaintiff argues that because Defendants are successors-in-interest to Hypercom, *CardSoft I* collaterally estops Defendants from re-litigating the construction of the “emulatable” term that *CardSoft I* construed. Dkt. No. 85 at 19. Alternatively, Plaintiff argues that although the specification discloses that any computer running a virtual machine will be able to run programs designed for that virtual machine, “there is nothing in the language of the claims, or in the common specification, that requires the virtual machine to be capable of running the same unmodified application on multiple different communication devices.” *Id.*

Defendants respond that *CardSoft I* did not resolve “what it means to be ‘incompatible’ in the context of the claim.” Dkt. No. 93 at 15. Defendants submit that construction is necessary because Plaintiff’s expert “alleged in *CardSoft I* that the emulation limitation may be satisfied by compiling source code for different target computers.” *Id.* at 16. Defendants argue that the specification explains that “absent the virtual machine, the application programs would necessarily have to be modified in order to account for the differences in the hardware or operating systems of the target device.” *Id.*

(2) Analysis

The Court has construed this disputed term as part of the term "emulatable in different computers having incompatible hardwares or operating systems," addressed above. Nonetheless, the parties dispute whether the constituent term "incompatible" requires construction, and the Court has a duty to resolve that dispute. *O2 Micro*, 521 F.3d at 1362-63.

Claim 1 of the '945 Patent is representative and recites, in relevant part:

> 1. A communication device which is arranged to process messages for communications, comprising a virtual machine means . . .
> . . .
> wherein the virtual machine means is emulatable in different computers having incompatible hardwares or operating systems.

The specification discloses:

> A virtual machine is computer programmed to emulate a hypothetical computer. *Different incompatible computers may be programmed to emulate the same hypothetical computer*. Any computer programmed to emulate the hypothetical computer will thus be capable of *executing programs for the virtual computer*. This creates a complete portable environment for program operations.

'945 Patent at 3:40-46 (emphasis added); *see id.* at 5:7-8 ("Each brand [of payment terminal device] is seen by the application as the same virtual machine."). The specification further discloses that in the absence of a virtual machine, application program instructions traditionally need to be modified or rewritten for different, incompatible devices. *See* '945 Patent at 9:45-54.

Defendants have also submitted an extrinsic dictionary definition from *Webster's New World Dictionary of Computer Terms*, which defines "compatibility" as:

> The capability of a device, program, or adapter to function with or substitute for another make and model of computer, device, or program. Also, the capability of one computer to run the software written to run on another computer. To be truly compatible, a program or device should operate on a given system without changes; all features should operate as intended and run, without changes, all the software the other computer can run.

Dkt. No. 93 at Ex. M. Although such extrinsic evidence must be used with caution, it is "among the many tools that can assist the court in determining the meaning of particular terminology to those of skill in the art of the invention." *Phillips*, 415 F.3d at 1318.

Under Defendants' proposed construction, however, two devices could not be deemed "incompatible" with each other if there could be found just one application that works on both devices without modifying the application and without using a virtual machine—even if perhaps thousands of other applications work on one of the devices but not the other. Thus, Defendants' proposal reads the constituent term "incompatible" too narrowly. Accordingly, the Court hereby rejects Defendants' proposed construction.

Instead, compatibility has its plain and ordinary meaning, albeit a meaning that is fact-sensitive and context-dependent. Because no additional "specificity and precision is warranted by the language of the claim and the evidence bearing on the proper construction, the task of determining whether the construed claim reads on the accused product is for the finder of fact." *PPG Indus.*, 156 F.3d at 1355.

The Court therefore hereby construes **"in different computers having incompatible hardwares or operating systems"** to have its **plain and ordinary meaning.**

CONCLUSION

The Court adopts the constructions set forth in this opinion for the disputed terms of the patents-in-suit.

The parties are ordered that they may not refer, directly or indirectly, to each other's claim construction positions in the presence of the jury. Likewise, the parties are ordered to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

SIGNED this 27th day of November, 2013.

Roy S. Payne

ROY S. PAYNE
UNITED STATES MAGISTRATE JUDGE

APPENDIX A

Term	Parties' Agreement
"message instruction means"	Construed based on [35 U.S.C. §] 112, Paragraph 6: (1) the function is "providing directions for operation of the virtual message processor;" and (2) the structure is described in the U.S. Patent No. 6,934,945 at "13:29-14:2; 15:23-34; Figure 11 and Figure 8, and equivalents thereof."
"virtual message processor"	"software implemented in the native code of the communications device that processes messages, including assembling, disassembling, and comparing messages, for communication to and/or from a communications device"
"assembling the messages, disassembling messages and comparing the messages"	Plain and ordinary meaning, does not require further construction.

Dkt. No. 60 at 2; Dkt. No. 98, Ex. A at 4-5 & 9-10.